

Tehisintellekt: sissejuhatus



Loengu teemad

- Orientatsioon
- Otsing
- Masinõpe
- Mõned näitesüsteemid ja valdkonnad

Orientatsioon

Marc Andreessen:

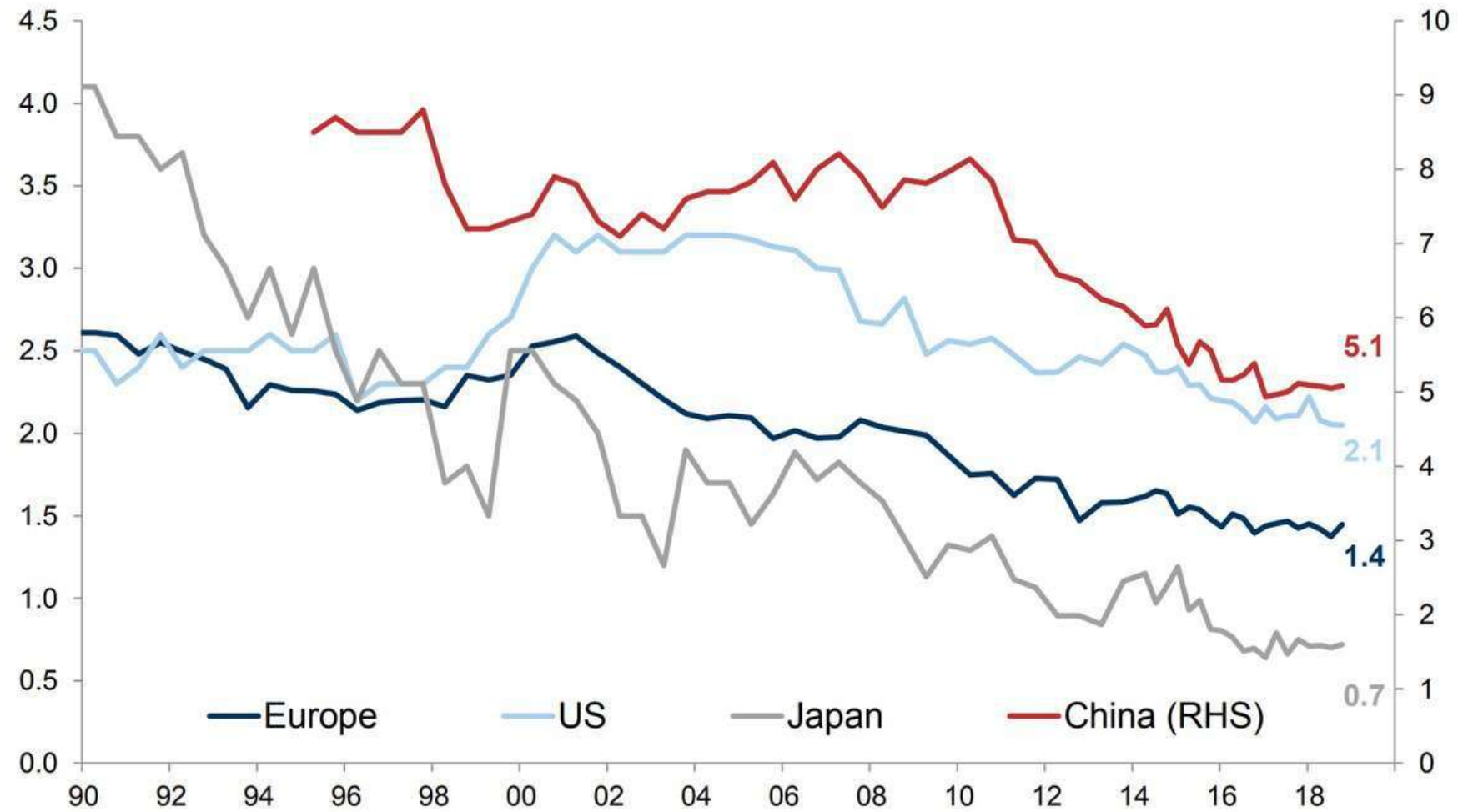
Software Is Eating the World

Paradoks:

IT ei ole sel sajandil globaalmajandust oluliselt efektiivistanud: vastupidi, GDP kasvutempo on pikaajalises langustrendis.

Exhibit 8: Long-term real global GDP growth forecast is at a historical low

Long-term (6-10y) GDP growth from Consensus Economics



Source: Consensus Economics, Goldman Sachs Global Investment Research

IT globaalne probleem ja lootus

Tööprotsesside automatiseerimine on väga raske: tarkvara ehitamine on aeglane ja kallis.

Tehismõistuse tehnoloogiad aitavad efektiivsemalt ehitada automatiseerimis-tarkvara.

Tugev ehk „päris“ AI, mida meil ei ole:

Masin on intellektilt inimloomusele lähedane või ületab seda, suudab täita tüüpilisi inimülesandeid, kasutada laia spektrit taustateadmisi ja omab mingil määral eneseteadvust.

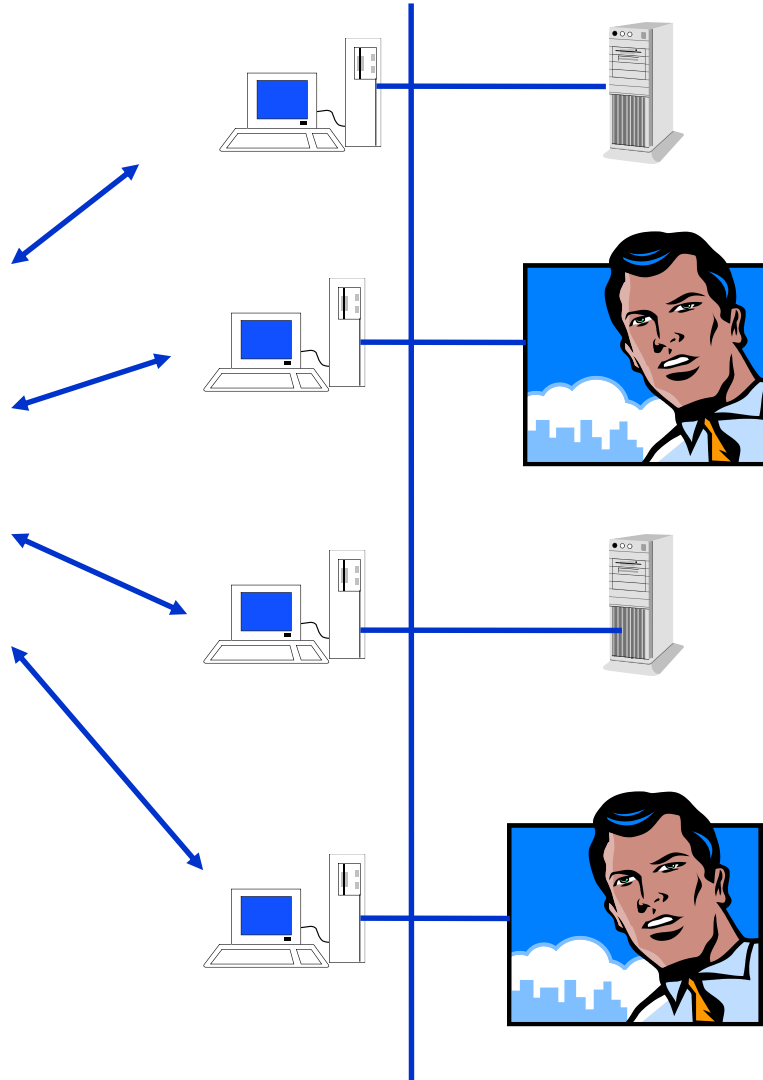
Nõrk AI ehk see, mis tegelikult on:

Tarkvara kasutamine spetsiifiliste probleemide või mõtlemisülesannete lahendamiseks, mis ei vaja laia spektrit inimvõimeid.

Turingi test: filosoofiline idee/ettepanek 50-datest



Mõistata, kas
chati-ekraani
taga on inimene
või programm?



Turing:

Kui katsetajad
ei suuda ära
arvata (st
ära-arvamise
sagedus on
50% ja 50%
eksitakse), siis
on jutlev
masin
päriselt
intelligentne.

AGI? Teadvus? Filosoofilised zombid?

AGI: **a**rtificial **g**eneral **i**ntelligence

Teadvus: endiselt müstiline värk 😊

Filosoofiline zombi: termin hüpoteetilise inimsarnase intelligentse olevuse kohta, kel aga ei ole üldse teadvust: n.ö. filosoofiline mõtte-eksperiment. Kellelgi pole aimdust, et kas nii saab olla, või ei saa olla, või mis on teadvus.



Moravec'i paradoks:

Inimeste jaoks rasked ülesanded on masina jaoks tihtipeale lihtsad.

Inimeste jaoks lihtsad ülesanded on masina jaoks tihtipeale äärmiselt keerulised.

Inimene ei teadvusta oma ajutegevust.

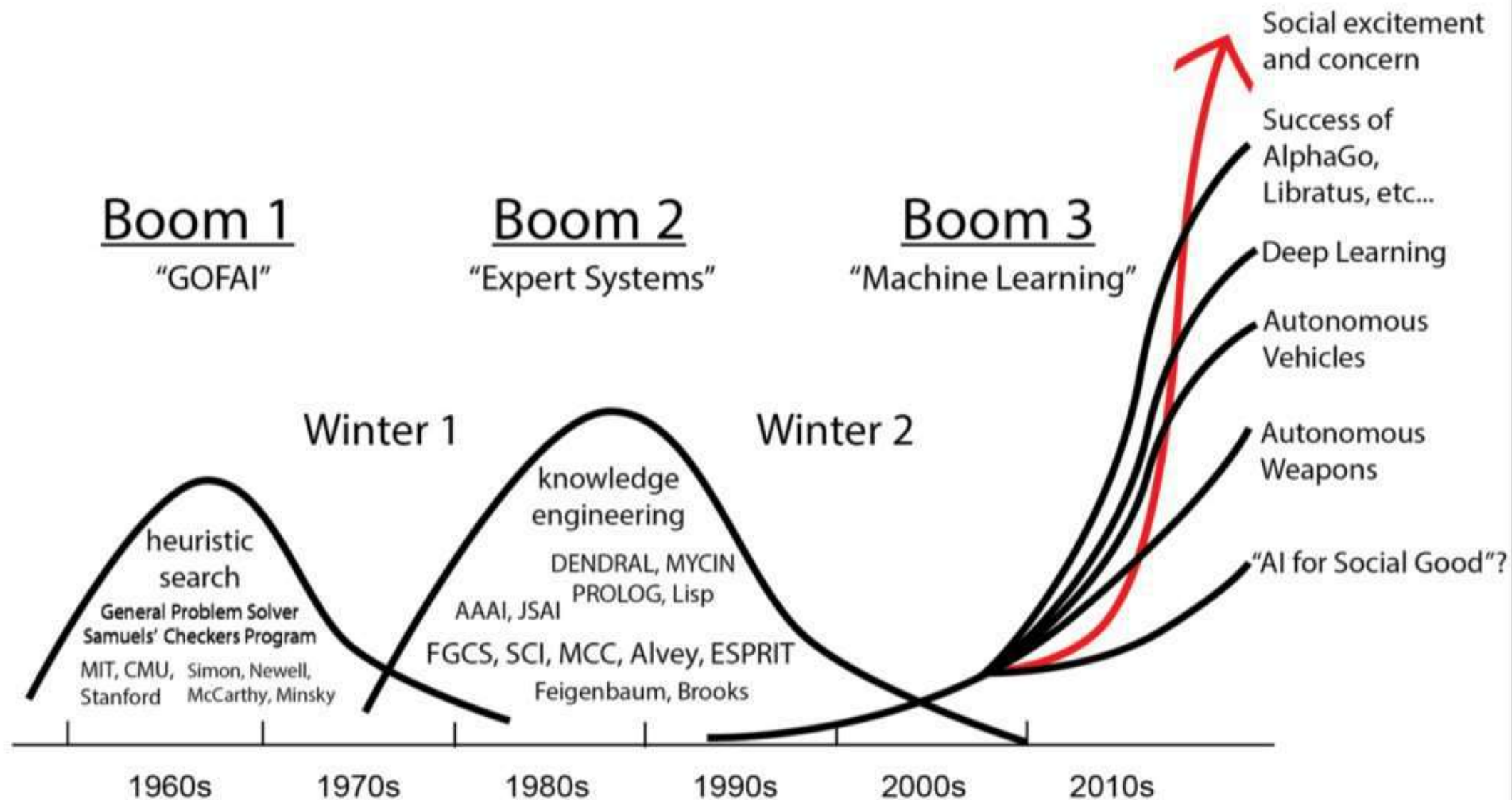
Inimesele paistab ekslikult, et paljud tema igapäevased toimingud on väga lihtsad ja ei nõua palju teadmisi ja ülikeerukaid ajuprotsesse

„Lihtne tehisintellekt“ ei ole võimalik?

Süsteemi võimsamaks tegemiseks on paratamatult vaja lisada üha rohkem ja rohkem erinevate spetsiaalsustega komponente?

Modern maxim:

We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run





Peamised A.I. meetodid

Kaks suurt AI meetodite valdkonda:

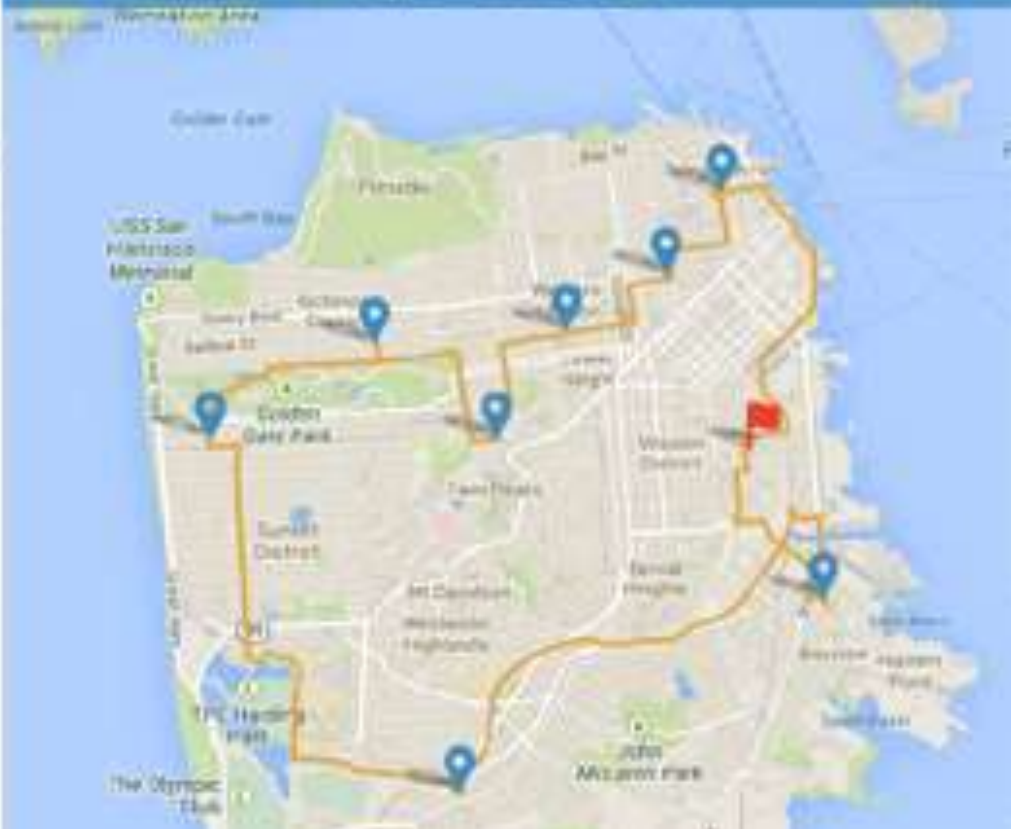
- **Otsing** paljude variantide seast a la malemäng.
- **Masinõpe** a la kasside ja koerte äratundmine pildilt, peale väga paljude õigesti sildistatud piltide näitamist.

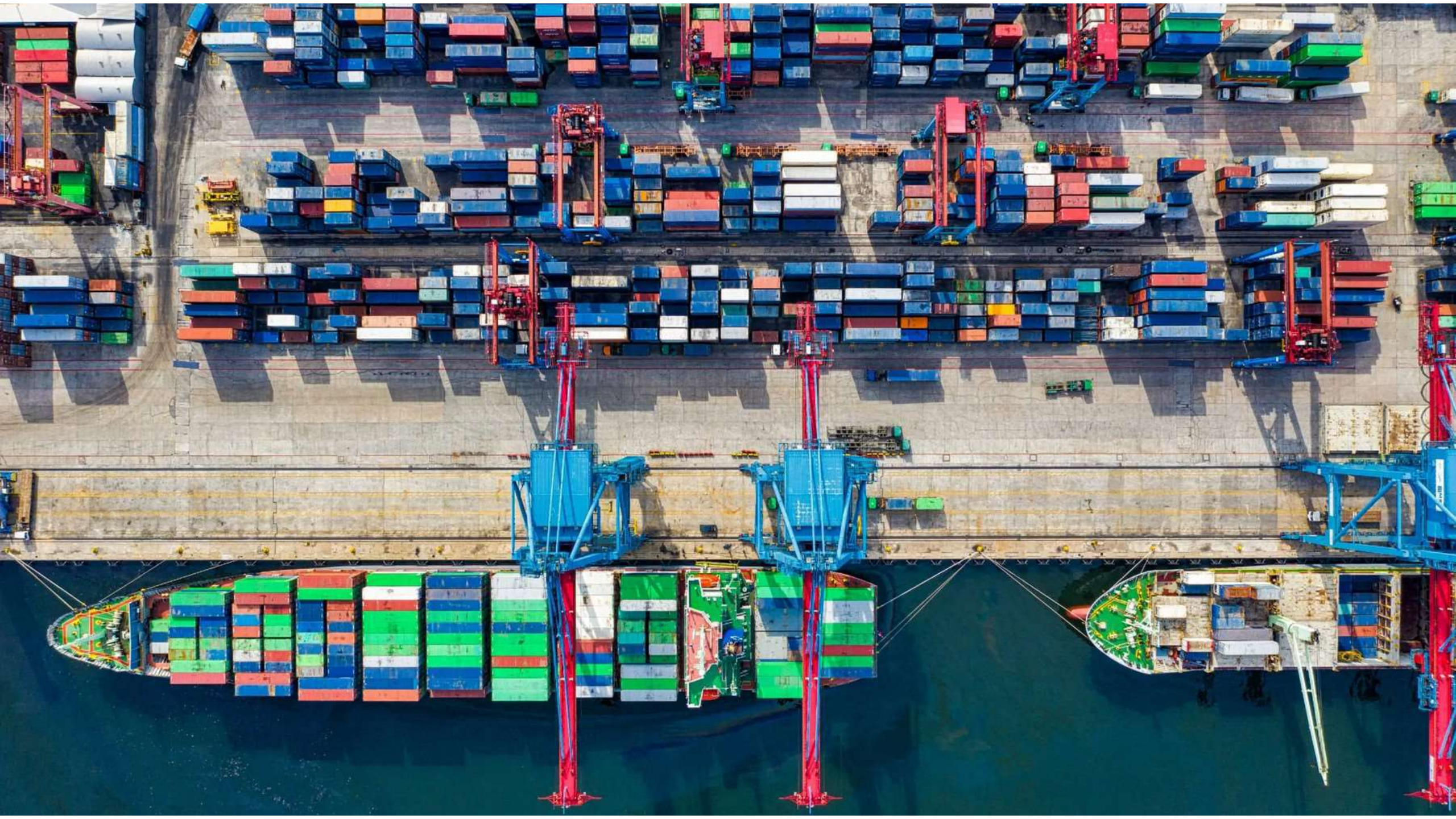
Otsing

Unoptimized Route

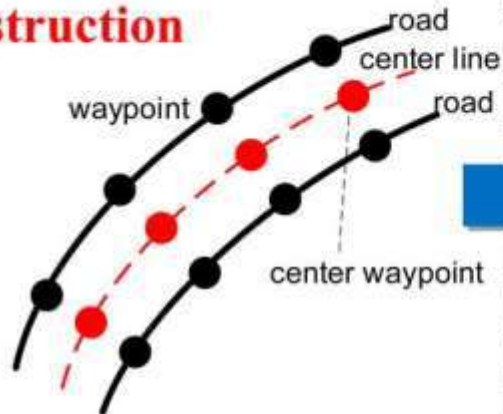


Optimized Route

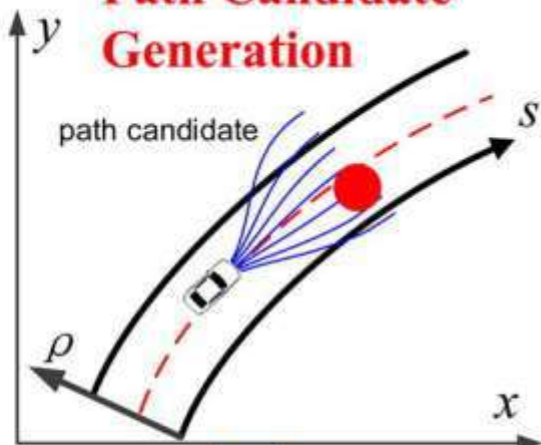




Center Line Construction

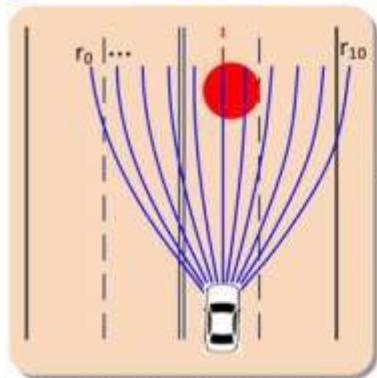


Path Candidate Generation

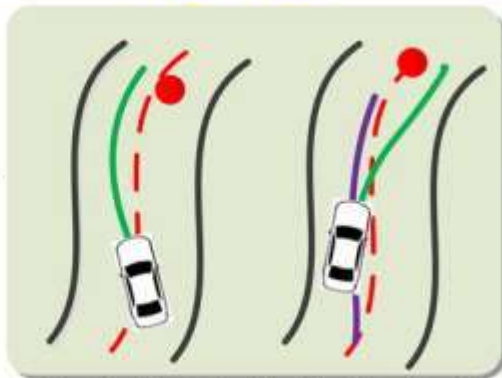


Path Selection

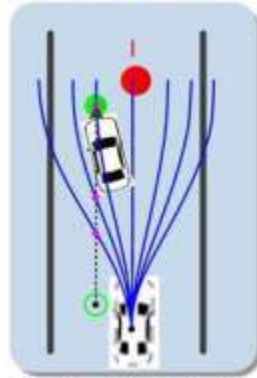
$$f = \omega_s f_s(r_i) + \omega_c f_c(r_i) + \omega_d f_d(r_i, a(r_i))$$



+

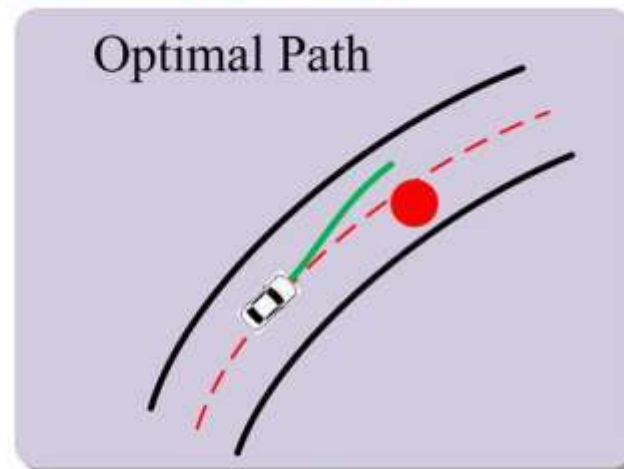


+

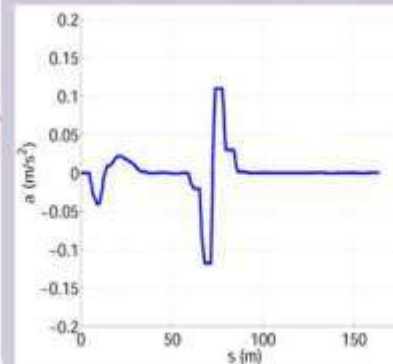


Path Planning Results

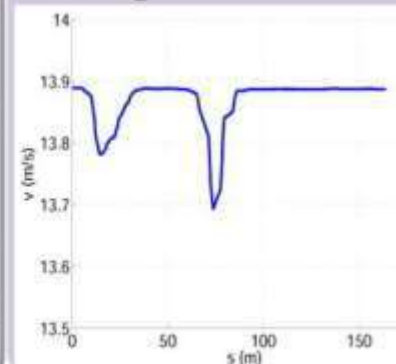
Optimal Path

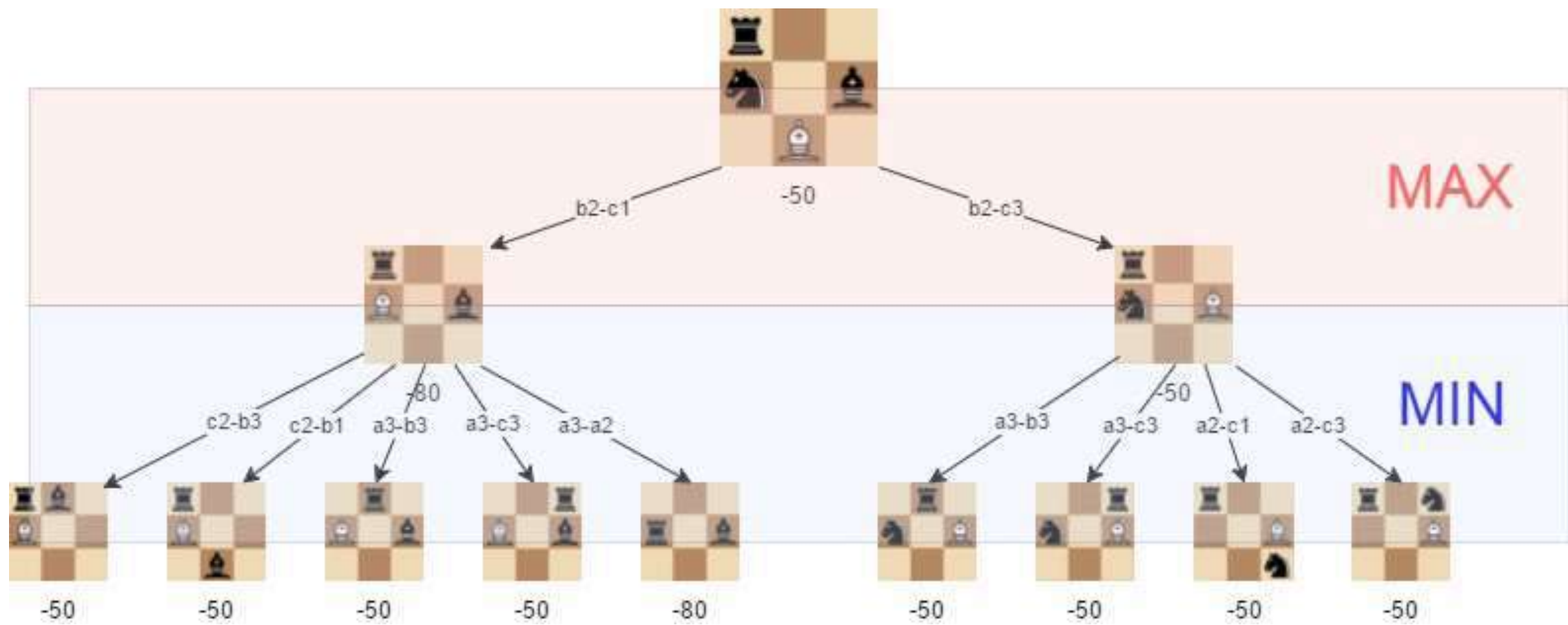


Appropriate Acceleration



Appropriate Speed





Kuidas maleprogrammi teha?

- Teeme kõigepealt programmi, mis:
 - loeb seisu
 - teeb seisu järgi mällu tabeli kõigist võimalikest käikudest selles seisus.
- See ei ole nii väga keeruline programm
 - Males keerulisem
 - Kabes lihtsam
 - Othello (reversi) veel lihtsam
 - Viis nuppu ritta: väga lihtne
- Nüüd võib programm tabelist juhusliku käigu valida.
- Ja juba mängibki! Kuigi kehvasti

Kuidas programmi veidi paremaks teha?

- Teeme eraldi väikese programmi, mis:
 - Võtab seisu ette
 - Ütleb umbes, kui hea seis on.
 - Seisu “headus” on number, mille meie programm arvutab.
- Kuidas öelda “kui hea”?
 - Vahel lihtne: kaotus / võit/ ei kumbki.
 - Numbrid: 1 -1 0
- Keerulisem: “harilik maleseis”. Kumbalgi pole võitu ega kaotust.

Seisu hindamine

- Materjal:

- Loeme kokku meie ja vastase nupud.
- Lipp annab 10 punkti, vanker 6, jne.

- Positsioon:

- Loeme kokku eelviimasel real etturid masinal/inimesel
- Loeme kokku tsentriväljadel olevad etturid masinal/inimesel
- Loeme kokku, mitmele tsentriväljale masin/inimene tuld annab
- Vaatame üle kuninga kaitse tugevuse
- jne

- Kokku saame seisu headuse numbri:

- Masina materjal – inimese materjal + masina positsioon – inimese positsioon

Mis edasi?

- Nüüd on meil kõigi käikude tabel, iga käigu juures arvutame tekkiva seisu headuse numbri.
- Valime kõige parema tulemuse (maksimumi numbritest!) andva käigu.
- Programm hakkab veidi paremini mängima, aga:
 - Ikka mängib väga kehvasti!
 - Miks? Sest meie seisu headuse hindamise programm on nigel.
 - Kuidas headuse hindamist parandada?
- Loomulikult tuleks vaadata, mis käike vastane võib peale meie käike teha! Ja kuidas meie võime vastata. Jne.

Kuidas sellist seisude puud kasutada?

- Eeldame lihtsalt, et:

- masin tahab teha käiku, mis annab kõige suurema headuse numbriga seisu.
- vastane tahab teha käiku, mis annab kõige väiksema headuse numbriga seisu.

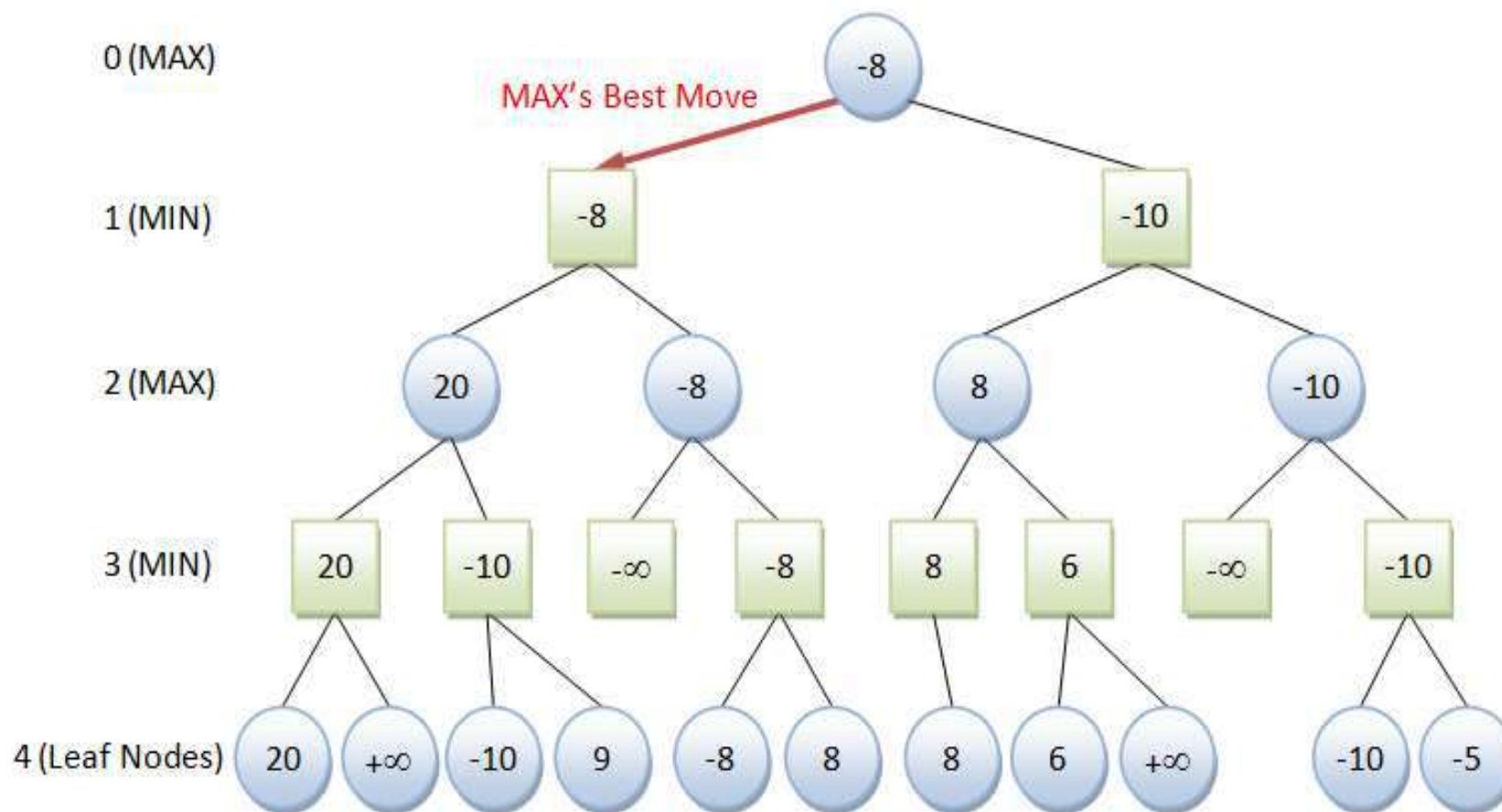
- Seega:

- igast seisust valib masin käigu, mis on maksimum-headusega (masinale).
- igast seisust valib vastane käigu, mis on miinimum-headusega (masinale).

- Idee:

- Vaatame käikude puud sügavuseni N (näiteks $N=3$)
- Kõige alumistel seisudel arvutame lihtsalt headuse välja
- Seejärel “tõstame” headuse numbreid ülespoole!

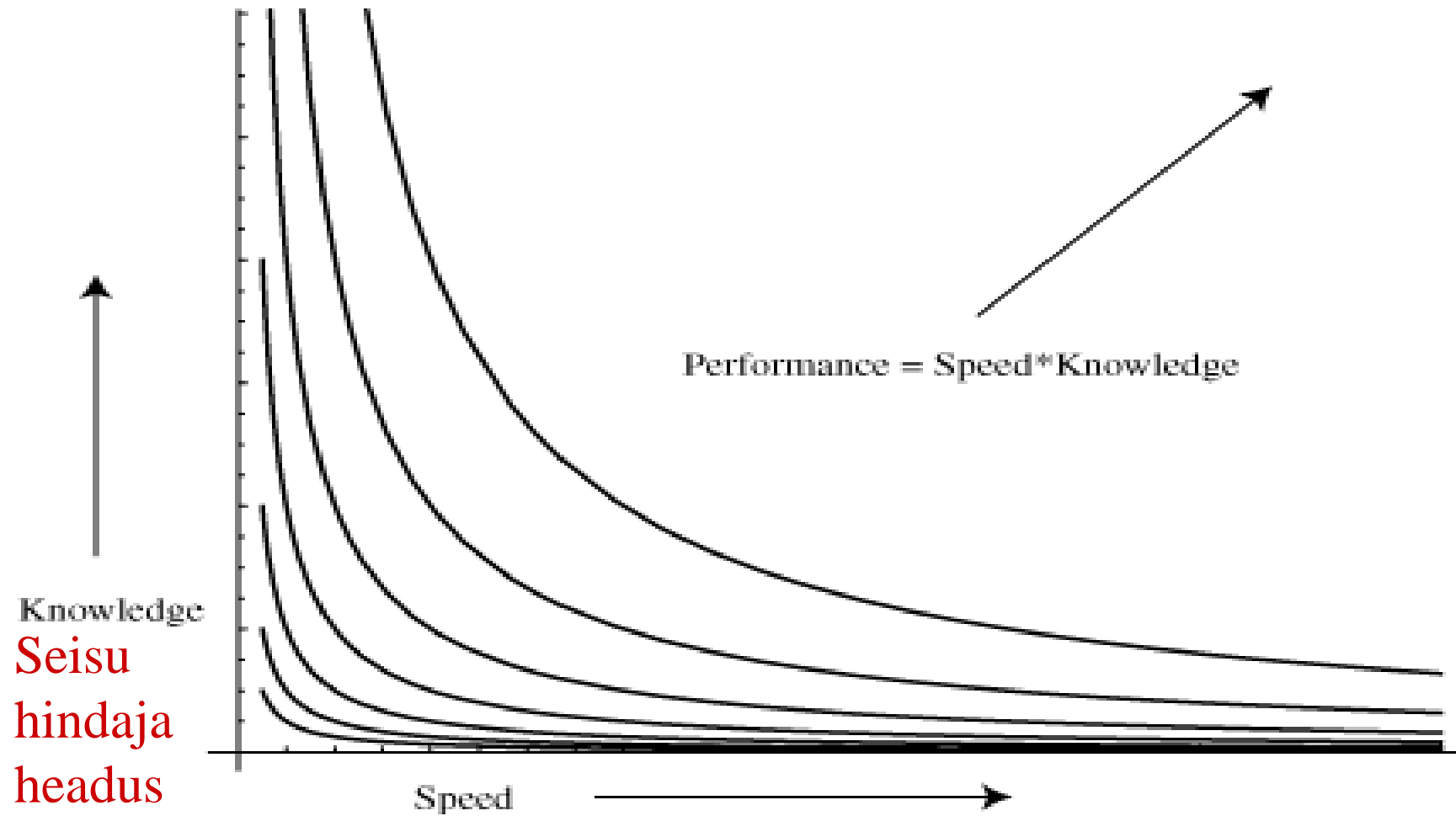
Otsing mängupuust: **minimax** algoritm



Kui heaks programmi saab?

- Mida sügavamad puud masin läbi jõuab vaadata, seda täpsemini ta käiku oskab valida.
- Puu läheb kiiresti väga suureks!
- Males ca 30 käiku ühes seisus.
 - Esimesel tasemel käike 30.
 - Teisel tasemel käike $30 \cdot 30$
 - Kolmandal tasemel käike $30 \cdot 30 \cdot 30$
 -
 - N-ndal tasemel käike 30 astmes N.
- Viiekümnnendal tasemel oleks käike ca 30 astmes 50. See on rohkem, kui elementaariosakesi universumis!

Kompromiss: teadmised ja kiirus ja mängu headus



Kui suure puu jõuab läbi vaadata

Kuidas programmi parandada?

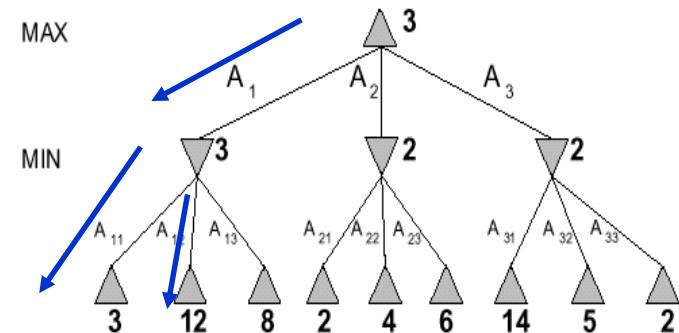
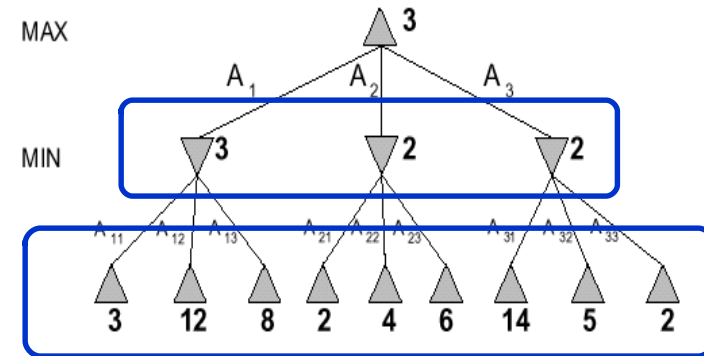
- Ei ole ühte head lahendust. On palju erinevaid nõkse!
- Näiteks:
 - Teeme seisu hindaja paremaks (programmi “targemaks”).
 - Aga siis läheb ta aeglasemaks ka.
 - Seega jõuame vähem käike läbi vaadata.
- Enamik nõkse on seotud käikude puu vähendamisega: ei ole vaja kogu puud läbi vaadata.

Laiuti vs sügavuti otsing

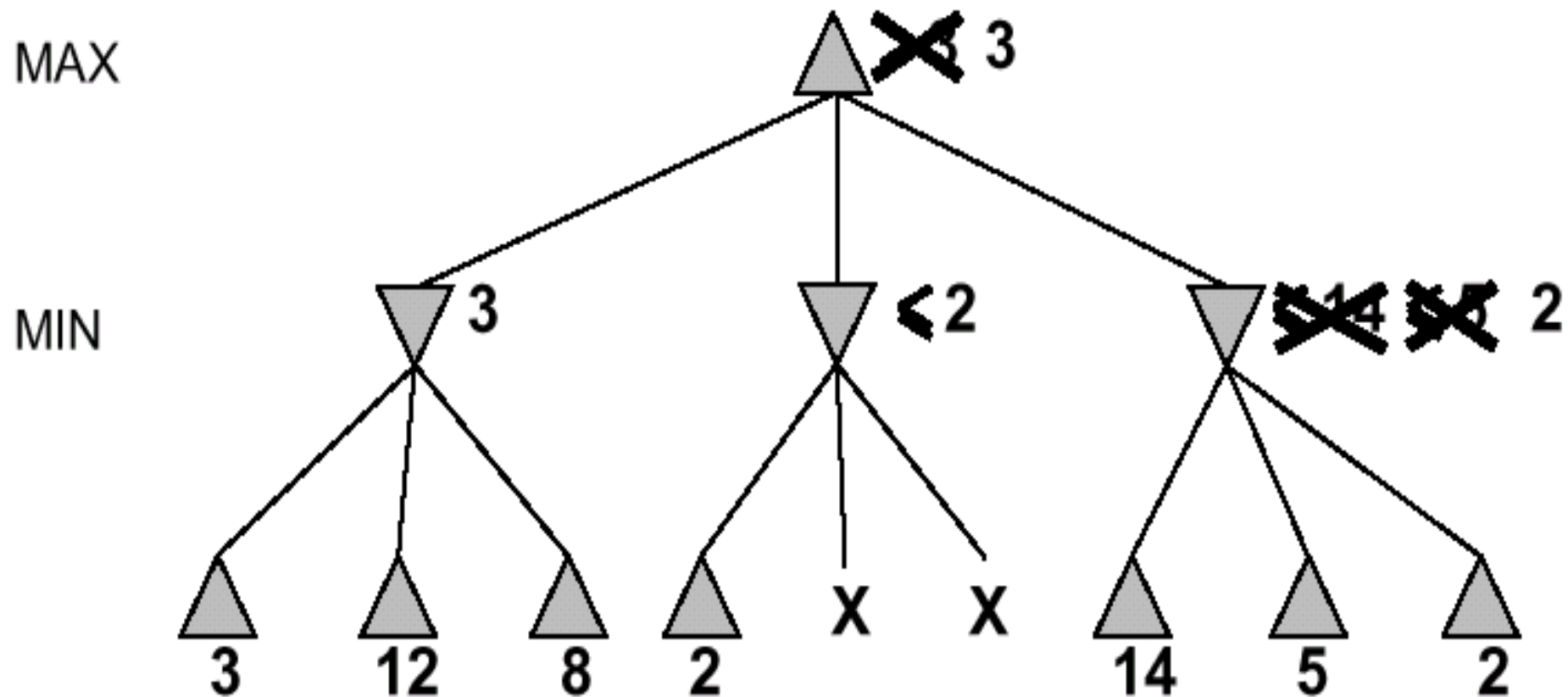
- Kaks võimalust:
 - Otsime puu läbi kiht-kihilt
 - Otsime puu läbi sügavuti, minnes algul vasakul maksimaalse sügavuseni

- Eelistatakse sügavuti otsingut!

- Mälu vaja palju vähem
- Muud eelised ka



Üks esimesi universaalseid meetodeid: alpha-beta



Sorteerimise nõks otsingu kiirendamiseks

- Sorteerida variandid seisust X enne otsingut ära: alustada tõenäoliselt paremate käikude proovimisega.
 - Suurendab tohutult alpha-beta efekti!
 - Kuidas sorteerida?
- Iteratiivne süvenemine. Teeme:
 - algul täisotsingu sügavuseni 2,
 - siis uue täisotsingu sügavuseni 4,
 - siis uue täisotsingu sügavuseni 6,
 - jne
 - iga kord kasutame eelmise otsingu tulemust sorteerimiseks!

Muud standardnõksud

- **“Killer moves”**: jätame otsides meelde eriti head käigud:
 - nii masinal kui vastasel
 - proovime kõigepealt varasemast meelde jäetud eriti häid käike
- **“Quiescence search”**: mõnda haru otsitakse sügavamalt:
 - ebastabiilses seisus otsime sügavamalt
 - stabiilses seisus otsime vähem sügavalt
 - otsime lõpuni kõik vahetused ja löögid
- **“Null-Move”**: mis siis, kui vastane saaks kaks käiku järjest?
 - Proovime nii, et vastane saab kaks käiku järjest
 - Kui on meile OK tulemus, siis see on positiivne faktor
 - Kui on meile halb tulemus, jätame meelde “killer move”

Kuidas optimeeringud mõjuvad?

■ Mida sügavam puu, seda suurem mõju. Näiteks:

■ Käikude puu sügavus viis:

- MiniMax: hindab 10,541,242 seisu
- Alpha-Beta: hindab 1,037,209 seisu
- A-B + “killer moves”: 530,587 seisu.

■ Käikude puu sügavus seitse:

- MiniMax: hindab ca 8,100,000,000 seisu
- Alpha-Beta: hindab 162,662,568 seisu
- A-B + “killer moves”: 46,455,262 seisu.

1997: Computer chess finally wins big

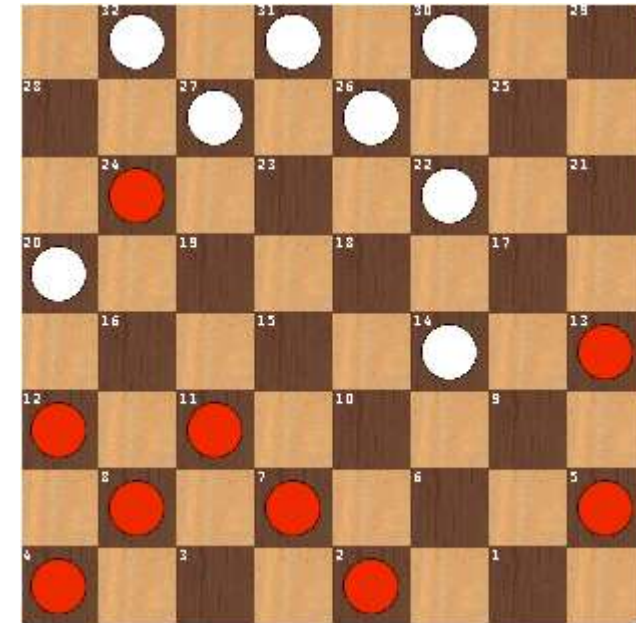
Deep Blue wins the reigning World Champion in Chess, Garri Kasparov



Inglise kabe lahendatud: 2007, Chinook UAlberta

Checkers has a search space of 5×10^{20}

Almost continuously since 1989, dozens of computers had been working around the clock to solve the game.



AlphaGo ja AlphaZero: hübriidsüsteemid

DeepMind / Google projekt masinõppeks Go ja male jaoks.

Iseenda vastu mängides õpib seisumustrite järgi seisu headust kiirelt ja hästi hindama.

Hea seisuhindamine on kombineeritud **Monte-Carlo otsinguga**: vaatab juhusliku valikuga läbi hulga võimalikke käikude jadasid mängu lõpuni välja.

GPT ja male (ehk male ilma otsinguta)?

Loe

<http://blog.mathieuacher.com/GPTsChessEloRatingLegalMoves/>

Otsimeetodite näide: ülesande lahendamine **loogikatõestuse** abil

- Ülesanne tuleb loogika keeles formuleerida.
- Küsimus tuleb ka loogika keeles formuleerida.
- Tõestaja asub tõestust otsima.
- Kui anda lõpmatult palju aega ja mälu, siis tõestaja lõpuks ka tõestuse leiab (kui tõestus üldse teoreetiliselt eksisteerib). Keeruliste ülesannete puhul võtaks see lootusetult kaua aega (meenuta keerukusklasse!)
- Kui tõestust ei ole, siis tõestaja enamasti jääbki seda otsima, teadma, et sellist tõestust ei saa olla.

Vana matemaatikaprobleemi lahendamine A.I. abil

W. McCune 1996:

- The Robbins problem---are all Robbins algebras Boolean?- has been solved using his automated theorem prover EQP.
- Programm otsis lahendust ca üks nädal, kuni lõpuks leidis.
- Ülesanne oli matemaatikute poolt lahendamata, kuigi püstitati aastal 1933:
- Meil on antud Robbinsi algebra:

$$\begin{array}{ll}x + y = y + x. & \text{[commutativity]} \\(x + y) + z = x + (y + z). & \text{[associativity]} \\n(n(x + y) + n(x + n(y))) = x. & \text{[Robbins equation]}\end{array}$$

- Kas järgmised võrdused annava selle algebra jaoks vastuolu:

$$\begin{array}{l}x+y \neq x. \\n(x+y) \neq n(x).\end{array}$$

Tõestuse algus

2 [] $x+y=y+x$.

3 [] $(x+y)+z=x+y+z$.

4 [] $(x+y)+z=x+y+z$.

5 [] $n(n(n(x)+y)+n(x+y))=y$.

6 [] $n(x+y) \neq n(x)$.

64 [para_into,4.1.1.1,2.1.1,demod,3] $x+y+z=y+x+z$.

71 [para_into,5.1.1.1.2.1,2.1.1] $n(n(n(x)+y)+n(y+x))=y$.

73 [para_into,5.1.1.1,2.1.1] $n(n(x+y)+n(n(x)+y))=y$.

75 [para_into,6.1.1.1,2.1.1] $n(x+y) \neq n(y)$.

93 [para_into,71.1.1.1,2.1.1] $n(n(x+y)+n(n(y)+x))=x$.

116 [para_into,75.1.1.1,4.1.1] $n(x+y+z) \neq n(z)$.

130 [para_into,93.1.1.1.2,73.1.1] $n(n(n(n(x)+y)+x+y)+y)=n(n(x)+y)$.

132 [para_into,93.1.1.1.2,5.1.1] $n(n(n(x+y)+n(x)+y)+y)=n(x+y)$.

139 [para_into,116.1.1.1.2,64.1.1] $n(x+y+z+u) \neq n(y+u)$.

170 [para_from,130.1.1,73.1.1.1.2,demod,3,3] $n(n(n(n(x)+y)+x+y+y)+n(n(x)+y))=y$.

174 [para_from,130.1.1,5.1.1.1.1,demod,3,3] $n(n(n(x)+y)+n(n(n(x)+y)+x+y+y))=y$.

190 [para_into,132.1.1.1.1.1.2,64.1.1] $n(n(n(x+y+z)+y+n(x)+z)+y+z)=n(x+y+z)$.

211 [para_into,139.1.1.1,64.1.1] $n(x+y+z+u) \neq n(x+u)$.

253 [para_from,170.1.1,93.1.1.1.2] $n(n(n(n(x)+y)+n(n(x)+y)+x+y+y)+y)=n(n(x)+y)$.

260 [para_from,170.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+z)+n(y+z))=z$.

346 [para_into,190.1.1.1.2,2.1.1] $n(n(n(x+y+z)+y+n(x)+z)+z+y)=n(x+y+z)$.

....

Tõestus jätkub

....

423 [para_from,253.1.1,5.1.1.1.1,demod,3,3,3,3] $n(n(n(x)+y)+n(n(n(x)+y)+n(n(x)+y)+x+y+y+y))=y$.
475 [para_from,260.1.1,73.1.1.1.2,demod,3,3] $n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+z+n(y+z))+z)=n(y+z)$.
486 [para_into,346.1.1.1.1.1.1.1,2.1.1,demod,3] $n(n(n(x+y+z)+x+n(z)+y)+y+x)=n(z+x+y)$.
684 [para_into,475.1.1.1.1.1.2.2,2.1.1] $n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+n(y+z)+z)+z)=n(y+z)$.
801 [para_from,684.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+n(y+z)+z)+z+u)+n(n(y+z)+u))=u$.
848 [para_into,801.1.1.1.1,486.1.1,demod,3] $n(n(x+x+n(n(x)+x)+x)+n(n(x+x)+n(n(x)+x)))=n(n(x)+x)$.
878 [para_into,848.1.1.1.2,93.1.1] $n(n(x+x+n(n(x)+x)+x)+x)=n(n(x)+x)$.
897 [para_into,878.1.1.1.1.1.2,64.1.1] $n(n(x+n(n(x)+x)+x+x)+x)=n(n(x)+x)$.
948 [para_into,897.1.1.1.1.1,64.1.1] $n(n(n(n(x)+x)+x+x+x)+x)=n(n(x)+x)$.
1015 [para_from,948.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(x)+x)+x+x+x)+x+y)+n(n(n(x)+x)+y))=y$.
1036 [para_from,948.1.1,73.1.1.1.2,demod,3,3,3] $n(n(n(n(x)+x)+x+x+x+x)+n(n(x)+x))=x$.
1053 [para_into,1015.1.1.1.2,423.1.1]
 $n(n(n(n(n(x)+x)+x+x+x)+x+n(n(n(x)+x)+n(n(x)+x)+x+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)$.
1079 [para_from,1036.1.1,93.1.1.1.2] $n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x)=n(n(x)+x)$.
1112 [para_into,1053.1.1.1.1.1,2.1.1,demod,3]
 $n(n(x+n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)$.
1130 [para_from,1079.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+y)+n(n(n(x)+x)+y))=y$.
1149 [para_into,1112.1.1.1.1.1,64.1.1]
 $n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)$.
1169 [para_into,1130.1.1.1.2,174.1.1] $n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+x+x+x)$.
1211 [para_into,1169.1.1,1149.1.1] $n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)=n(n(n(x)+x)+x+x+x)$.
1212 [binary,1211.1,211.1] \$F.

Mis on **ekspertsüsteem**

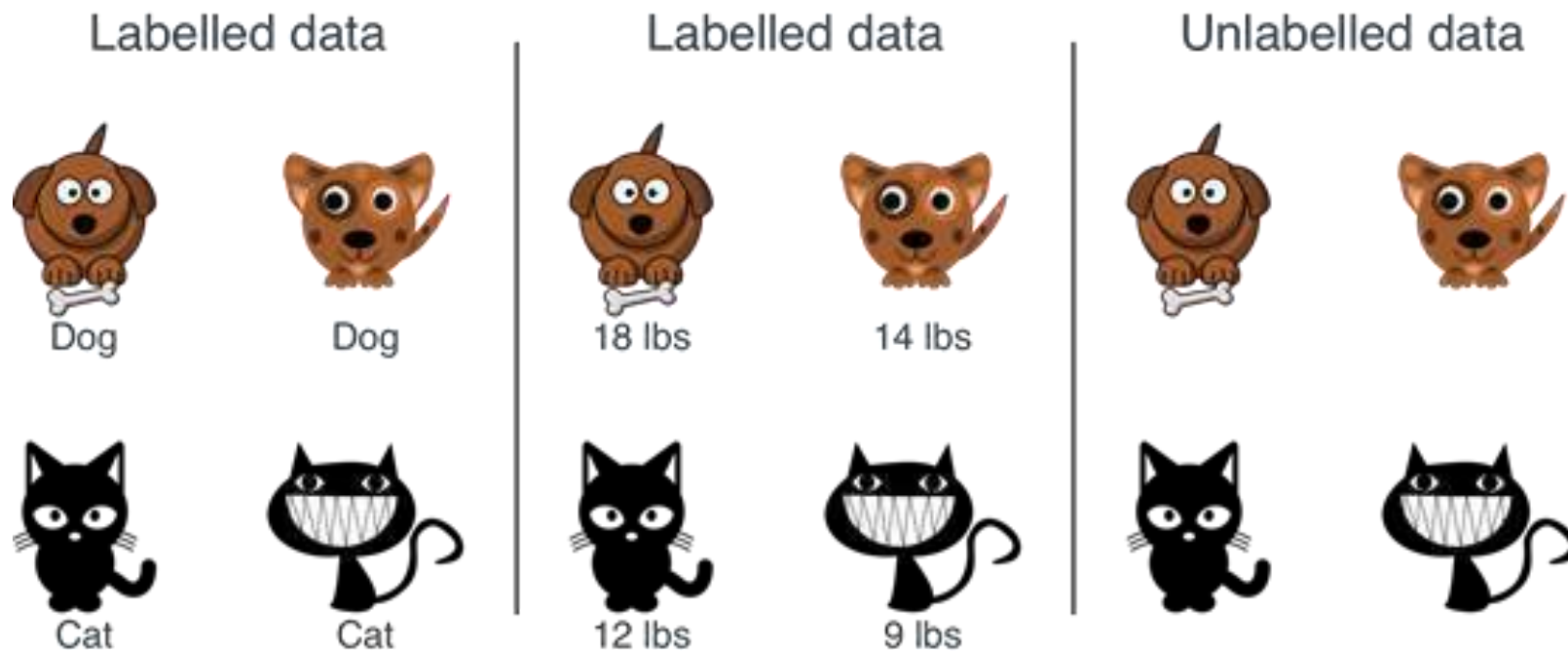
- Ekspertsüsteem on mõne spetsiifilise valdkonna jaoks kohandatud järelduste tegemise programm (näiteks variant teoreemitõestajast)
- Tihtipeale kasutavad ebaharilikku, spetsiaalselt valdkonnaga sobitatud loogikat.
- Enamasti sisaldavad paljusid valmiskujul reegleid.
- Enamasti sisaldavad mugavat kasutajaliidest: kasutaja ei pea loogika keelt oskama.
- Tüüpiliselt kasutatakse mõne suure rakenduse osana

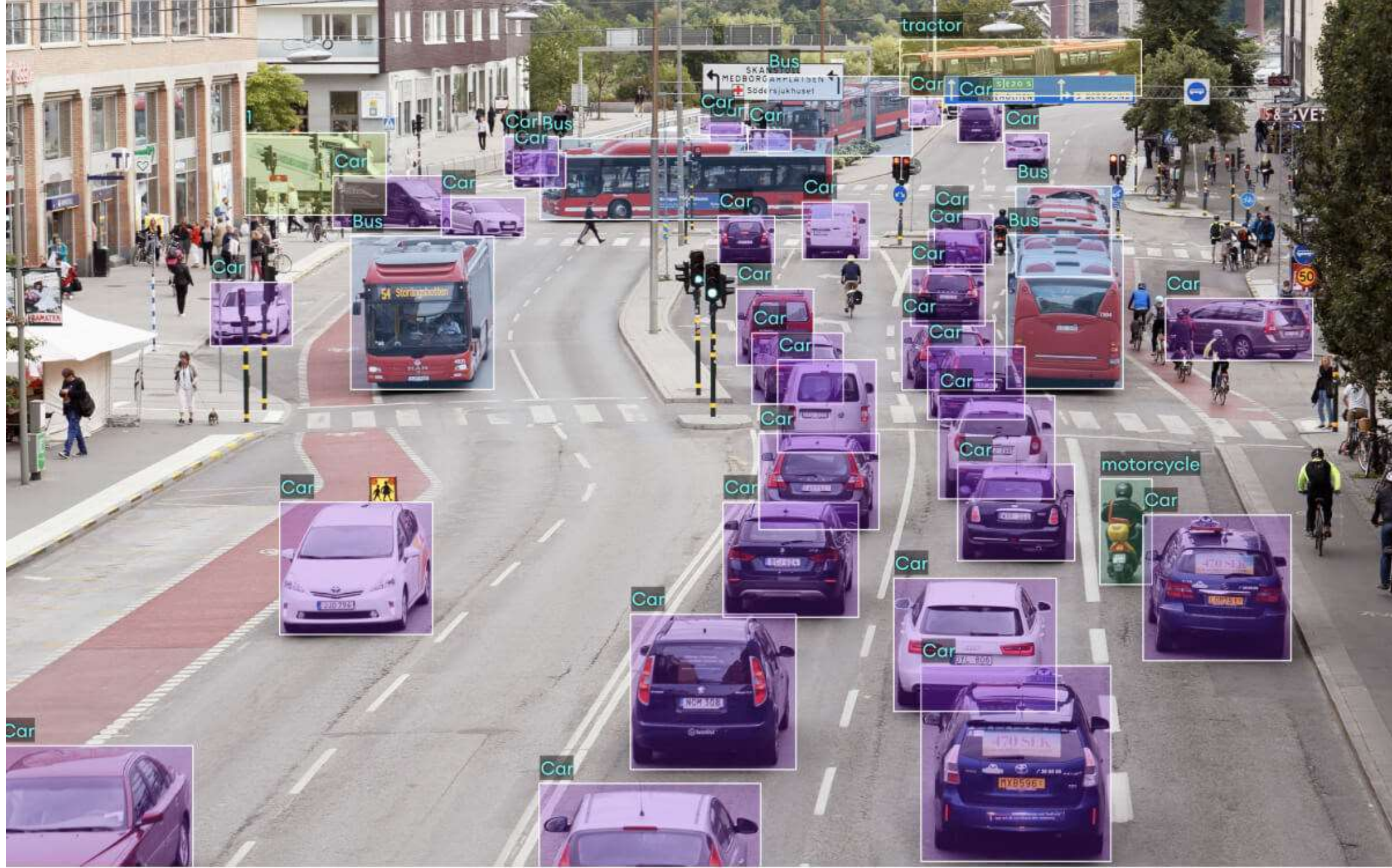
Masinõpe

Masinõppe kaks peamist viisi

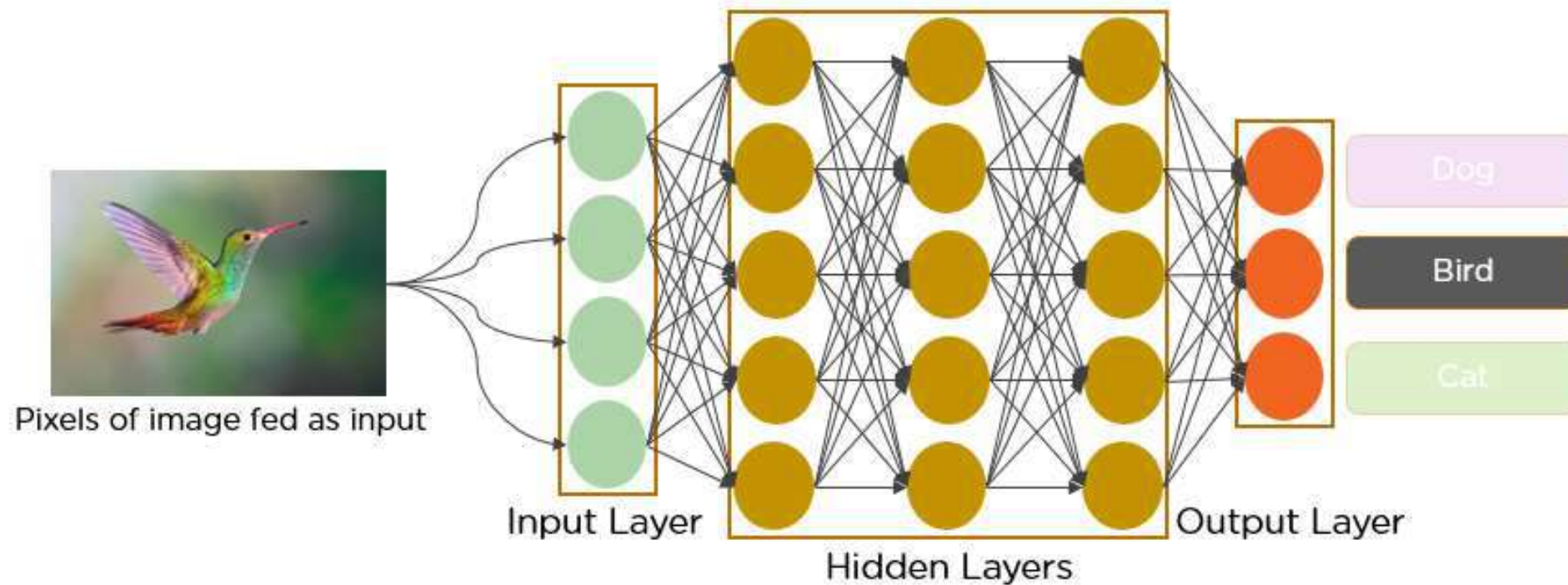
- **Sildistatud andmete pealt õppimine** (kassi pildid ja koera pildid koos inimese pandud sildiga: kumb on)
- **Eksperimentidega õppimine** (ise katsetad väga palju eri tegevusi ja kui saad positiivset tagasisidet, “jätad meelde”, et see oli ses olukorras tark tegu).

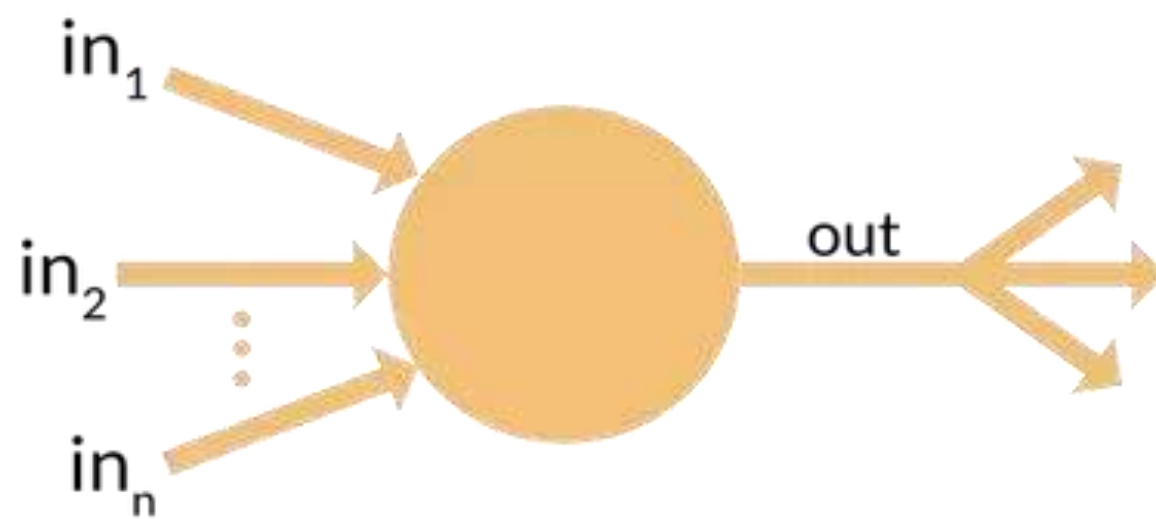
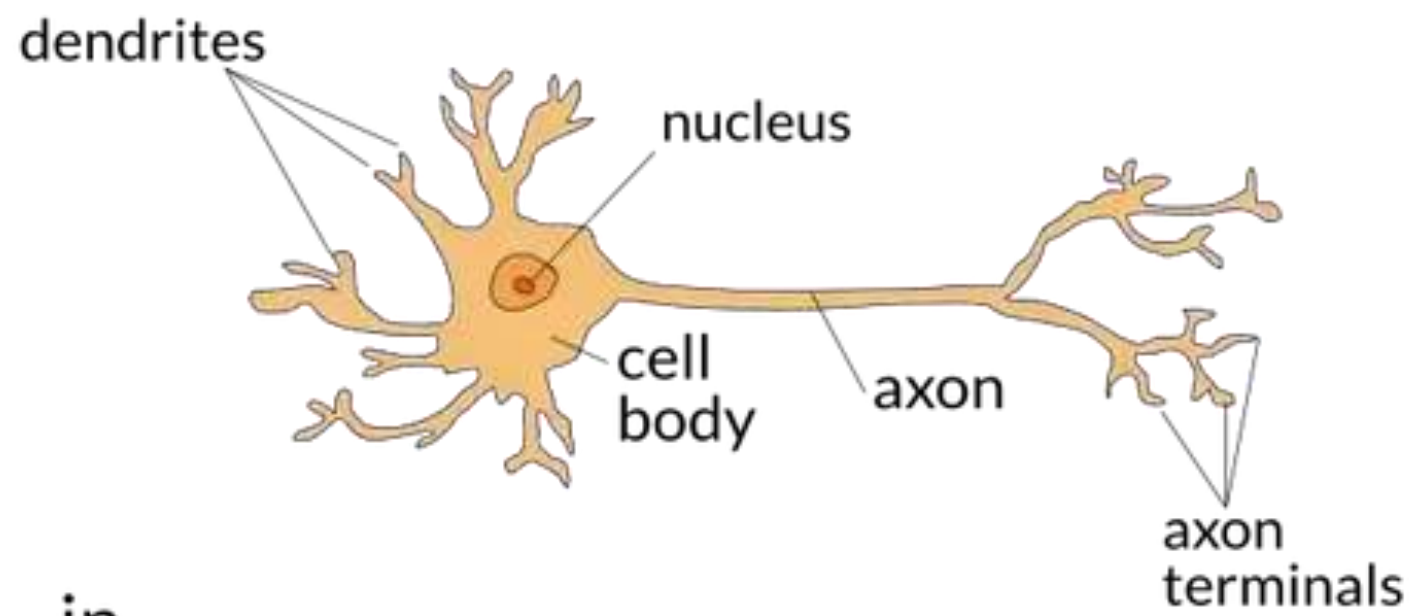
Sildistatud ja sildistamata andmed

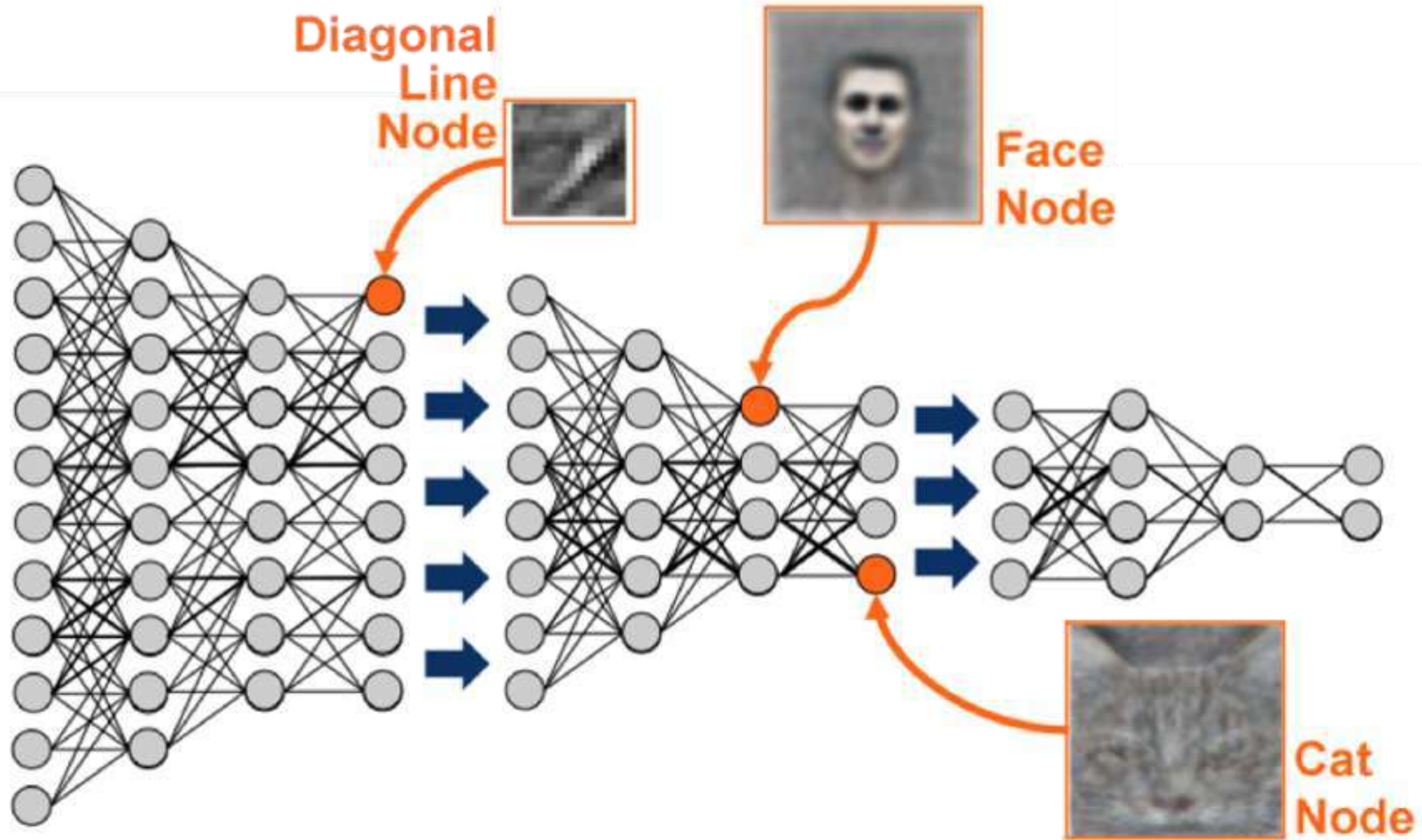




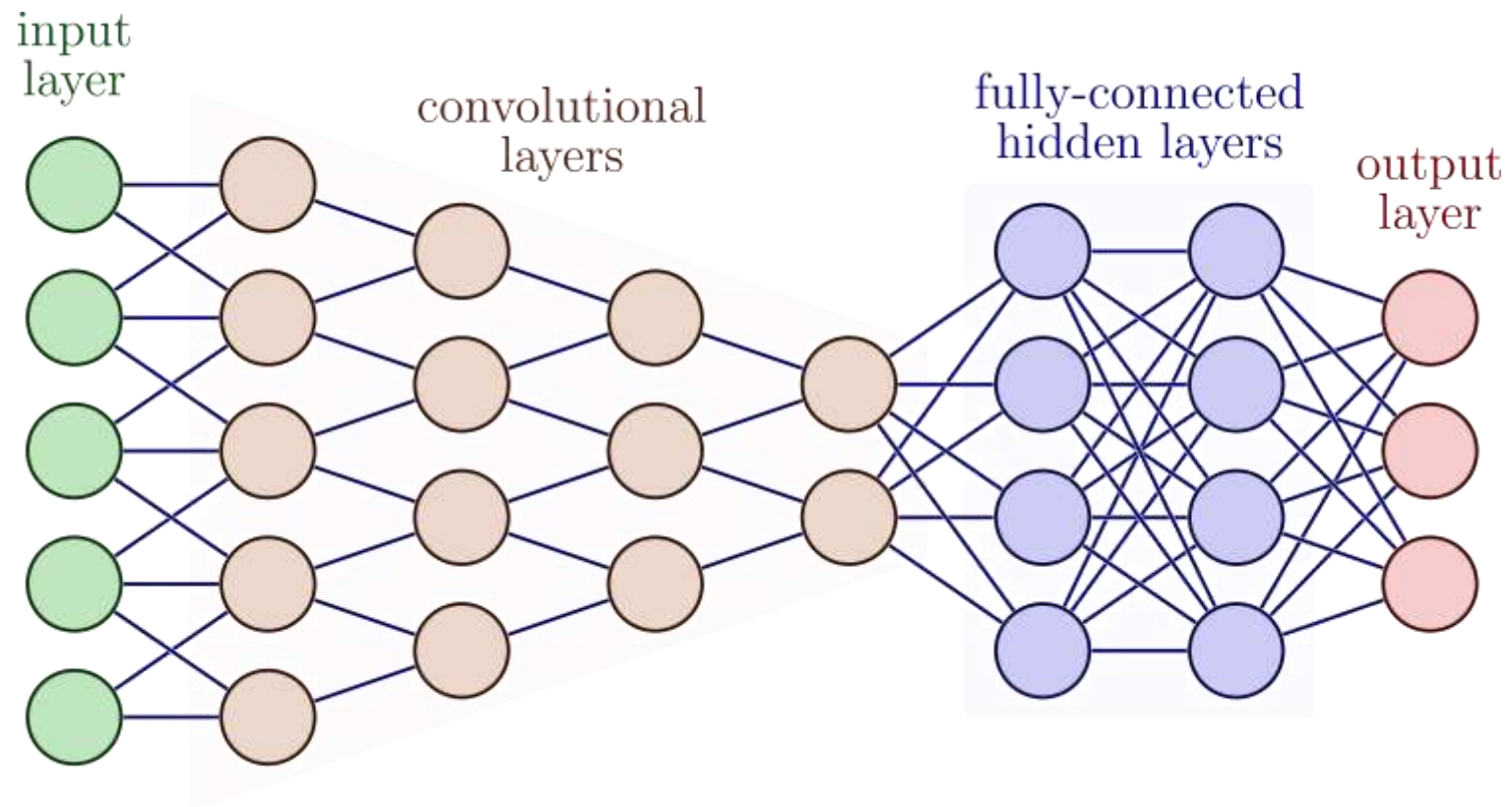
Deep neural networks ehk sügavad närvivõrgud



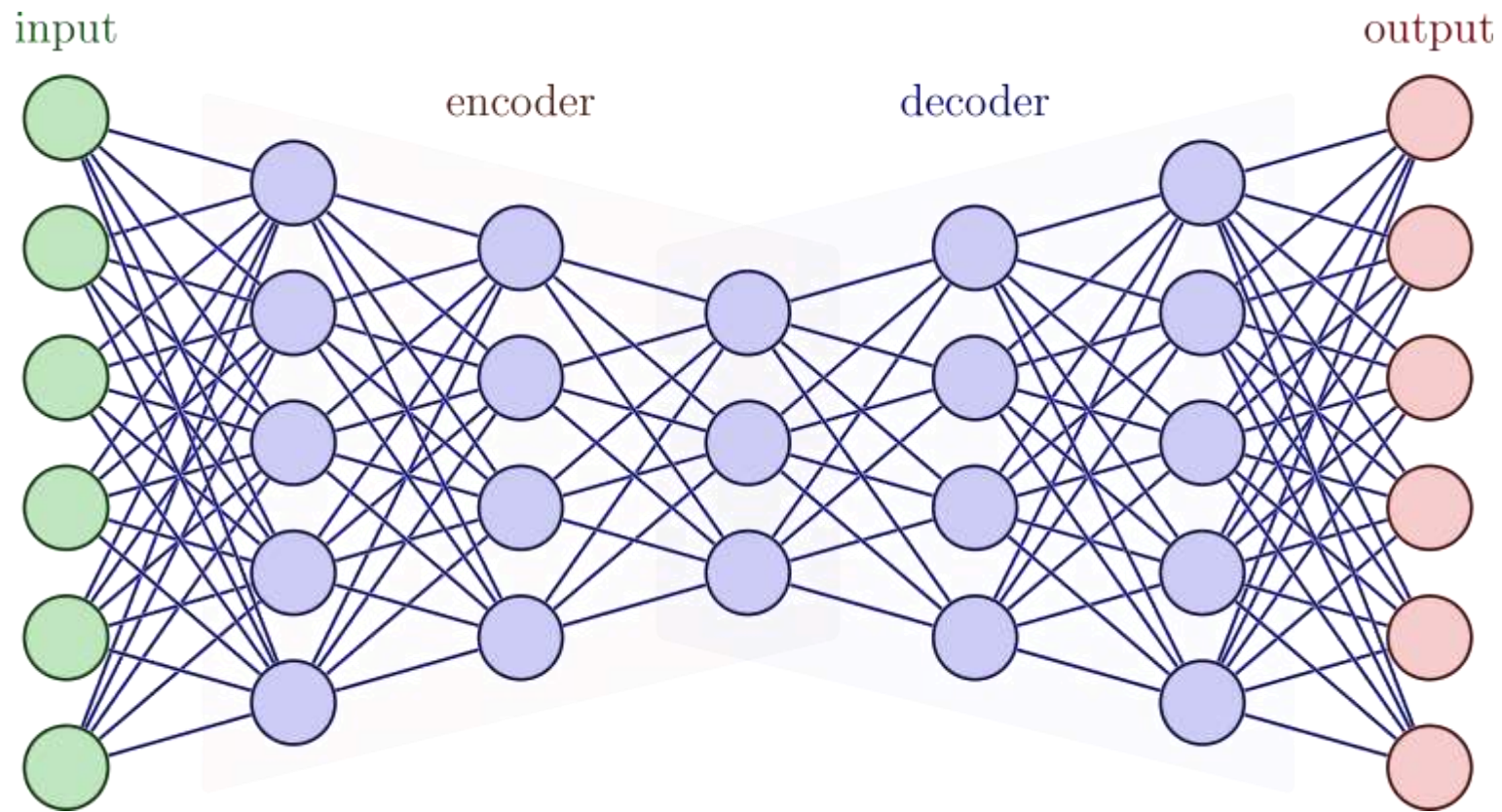




Convolutional neural network (image recognition)



Autoencoder neural network (denoising, compression, generation)



(a)



school bus 1.0

(b)



garbage truck 0.99

(c)



punching bag 1.0

(d)



snowplow 0.92



motor scooter 0.99



parachute 1.0



bobsled 1.0



parachute 0.54



fire truck 0.99



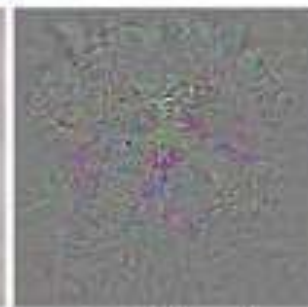
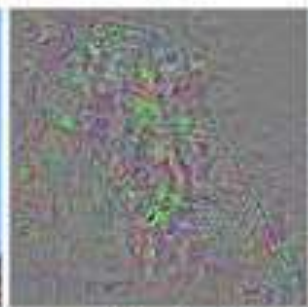
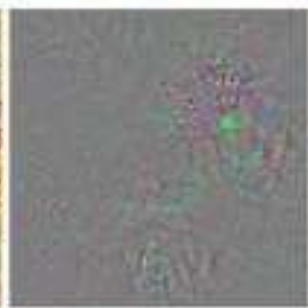
school bus 0.98



fireboat 0.98



bobsled 0.79



correct

+distort

ostrich


correct

+distort

ostrich

Näitesüsteemid, rakendused, valdkonnad

Wolfram alpha online: 2009

 **WolframAlpha**[™] computational knowledge engine

meaning of life

Assuming "meaning of life" is a quantity | Use as referring to English words or a movie instead

Input interpretation:
answer to life, the universe, and everything

Result:
42
(according to Douglas Adams' Hitchhiker's Guide to the Galaxy)

Computed by: [Wolfram|Mathematica](#) | Download as: [PDF](#) | [Live Mathematica](#)

IBM Watson võitis Jeopardy! aastal 2011



- It's just a bloody nose! You don't have this hereditary disorder once endemic to European royalty : *Haemophilia*
- You just need a nap! You don't have this sleep disorder that can make sufferers nod off while standing up : *Narcolepsy*
- Heitor Villa-Lobos dedicated his "12 Etudes" for this instrument to Andres Segovia : *Guitar*
- Paganini's "24 Capricci" set the standard for etudes for this instrument : *Violin*
- Rembrandt's Biblical Scene "Storm on the Sea of" this was stolen from a Boston museum in 1990 : *Galilee*

Masinõpe ja inimkeel: watsoni-aegsed tipp-süsteemid ei saanud hakkama selliste ülesannetega:

- Linnavalitsus ei andnud meeleavaldajatele kogunemiseks luba, sest nad kartsid vägivalda. Kes kartis vägivalda?
- Palgil on kuus konna. Kaks lähevad ära ja kolm uut tulevad kohale. Mitu konna on nüüd palgil?
- Ma viisin eile oma riided pesumajja ja pean neile varsti järele minema. Kus mu riided praegu on?

*Keeletehnoloogia tehismõistuse edulugu:
närvivõrgud ja nn Large Language Models (LLM)*

Google: BERT, LaMDA

Facebook: XLM-RoBERTa, LLaMA

OpenAI (Microsoft?): GPT2, GPT3, ChatGPT, GPT4

Nvidia: Megtron-Turing

....

Mida LLM teeb? Ennustab samm-sammult teksti jätkumist:

- Jaan istus autosse ja ...
- Jaan istus autosse ja hakkas ...
- Jaan istus autosse ja hakkas sõitma ...
- Jaan istus autosse ja hakkas sõitma kodu ...

Kuidas LLM ennustab?

- Kogub statistikat suure tekstihulga pealt: mismoodi seal on laused jätkunud, näiteks

→ Mihkel istus mootorrattale ja ... (*hakkas, mõtles, ...*)

- Kui näeb lauset, mille kohta statistikat ei ole, kasutab statistikat varasemate sarnaste lausete kohta:

Jaan istus autosse ja ...

Sarnane!

Miks on LLMi treenimine edukas?

Tekst ise ongi “sildistatud” näide, et kuidas ta peaks jätkuma!

- a) Kogud statistikat suure tekstikoguse kohta ja sätid võrguseoste numbreid
- b) Vaatad, kui hästi ennustus järgmiste lausete jaoks klapib tegeliku lausega
- c) Tuunid õppimisparameetreid ja kordad ja kordad ja kordad ...

LLMi põhisuutlikkused

- Ahvimine: genereerib sobivas stiilis teksti, mis on suhteliselt mõistlik
- Suudab teha mõõdukaid üldistusi ja lihtsamaid järeldusi

LLMi probleemid

- Hallutsineerib ehk leiutab “fakte”, mis ei ole üldse tõesed.
- Üldjuhul ei suuda edukalt seletada oma vastuseid / järeldusi
- Enamasti ei suuda teha pikemaid järeldusahelaid
- Ei ole lihtsat viisi siduda andmebaasidega, inseneriarvutustega jne
- Vastuste kvaliteet langeb kiirelt, kui sisendtekst on pikk
- Ei ole selget viisi tema suunamiseks või mingit sorti vastuste piiramiseks / keelamiseks

*ChatGPT saab kenasti hakkama nende
lihtsate näidetega:*

If an animal likes honey, then it is probably a bear.
Most bears are big, although young bears are not big.
John is an animal who likes honey.
Mike is a young bear.

Who is big? (John)

The length of the red car is 4 meters.
The length of the black car is 5 meters.

The length of the red car is less than 5 meters? (Yes)

Aga ChatGPT ei saa hakkama modifikatsiooniga

If a greezer likes foozers, then it is probably a drimm.

Greezers can eat frozen bread.

Most drimms are red, although young drimms are not red.

John likes bread.

John is a nice greezer who likes foozers.

Mike is a young drimm.

Mike can eat a lot. Penguins are birds who cannot fly.

Who is red?

... ja samuti selle modifikatsiooniga

The length of the barner is 200000000 meters.

The length of the red foozer is 312435 meters.

The length of the black foozer is 512000 meters.

The length of the yellow foozer is 1000000 meters.

The length of the red foozer is less than 312546 meters?

Robotika

Süntees kõigist A.I. valdkondadest, pluss
mehaanika

Iseõitvad autod

2005

Stanfordi Stanley
võidab DARPA
Grand Challenge II



Darpa urban challenge 2007

96 km linnas, 4:10

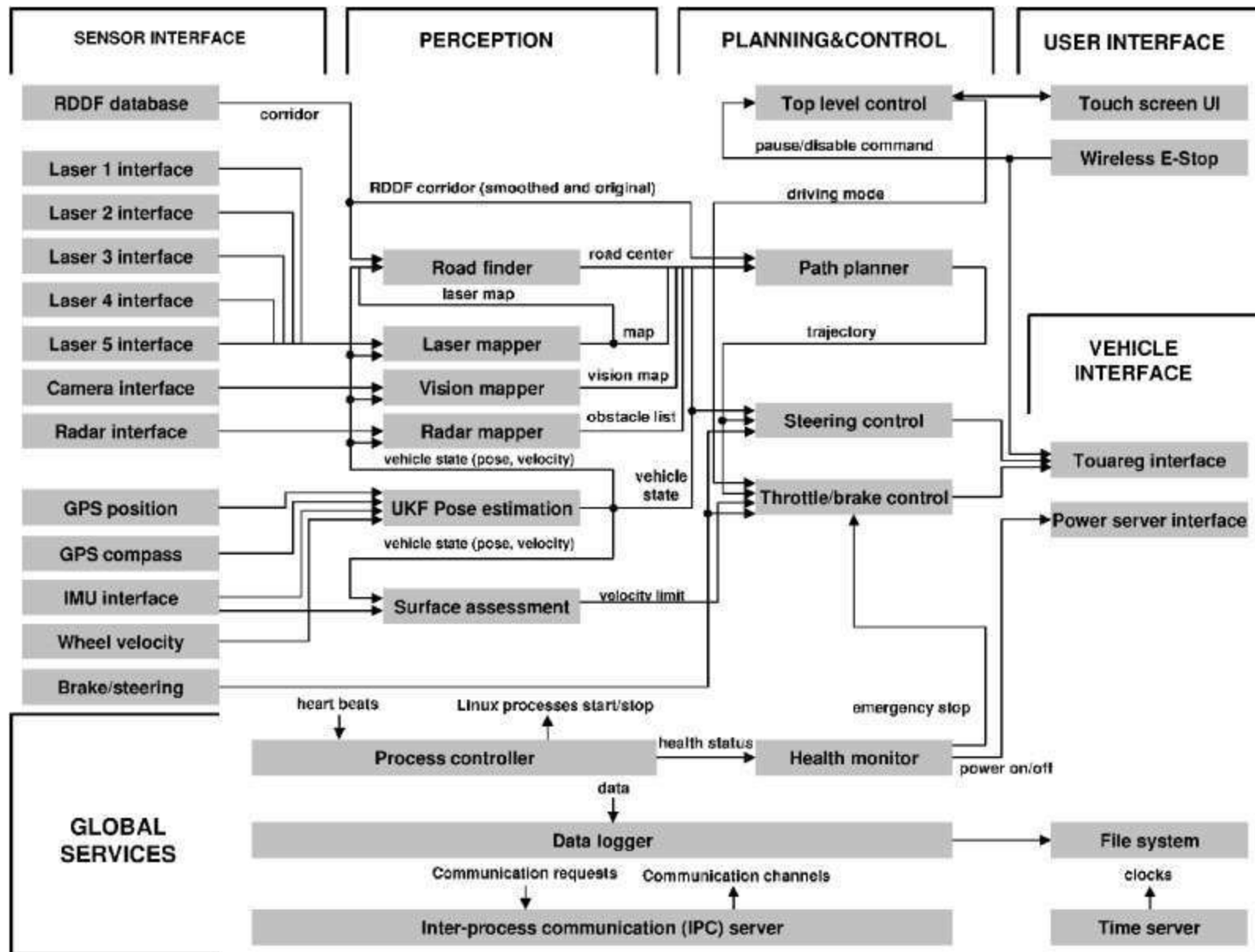


Google autonomous car

Based on vehicle Stanley which won the 2005 DARPA Grand Challenge.

In August 2012, the team announced that they have completed over 300,000 autonomous-driving miles (500 000 km) accident-free





Iseõitvad autod on keerulisemad, kui algul paistis

Chris Urmson, the former leader of Google's self-driving car project, once hoped that his son wouldn't need a driver's license because driverless cars would be so plentiful by 2020.

Now the CEO of the self-driving startup Aurora, Urmson says that driverless cars **„will be slowly integrated onto our roads over the next 30 to 50 years“**.

Tesla

„Traffic-Aware Cruise Control **cannot detect all objects and may not brake/decelerate for stationary vehicles**, especially in situations when you are driving over 50 mph (80 km/h) and a vehicle you are following moves out of your driving path and a stationary vehicle or object is in front of you instead.”

Volvo

„In other words, **your Volvo won't brake to avoid hitting a stopped car that suddenly appears up ahead**“.

iRobot
Roomba
algus 2002
käivitaja
Rodney Brooks

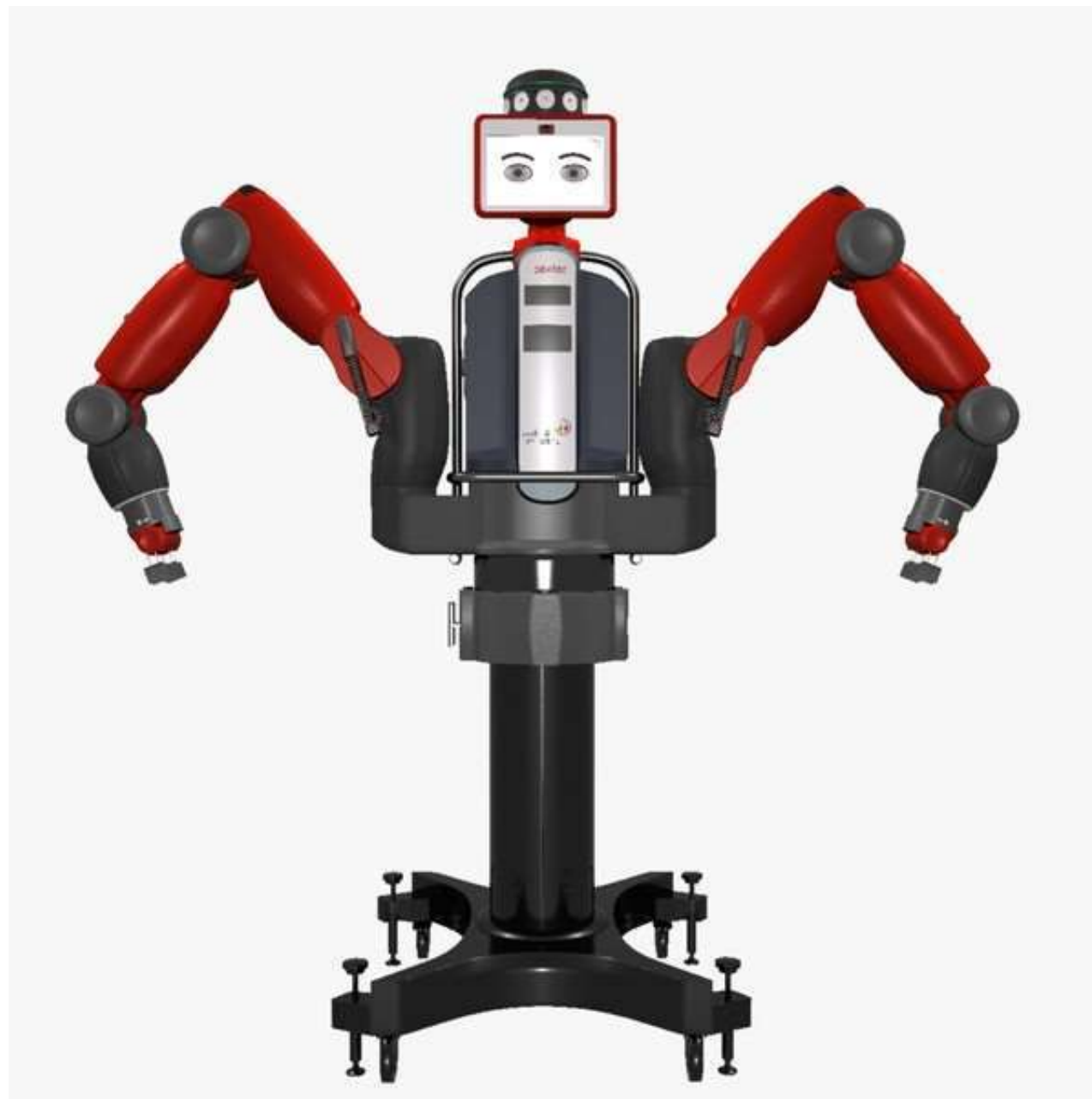


Rethink
Robotics

algus 2008

käivitaja
Rodney Brooks

lõpetas 2018



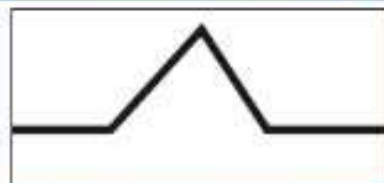
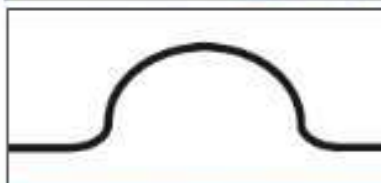
Rodney Brooks:

„Despite some impressive lab demonstrations we have not actually seen any improvement in widely deployed robotic hands or end effectors in the last 40 years“

A robot that can carry out the last 10 yards of delivery, getting from a vehicle into a house and putting the package inside the front door?

Prediction:

- Lab demo: 2025
- Deployed systems: 2028





Explainable A.I. ehk selgitatav tehismõistus

Masinõppe süsteemide tehtud otsuste taga on suur hägune võrgustik seoseid, mitte inim-arusaadavad reeglid.

Otsuste põhjendamiseks oleks lisaks hägusele võrgustikule vaja ehitada mõistetavad reeglid, mille abil otsust põhjendada.

Hetkel puhtalt akadeemiline uurimistöö, praktilisi süsteeme ei osata teha.

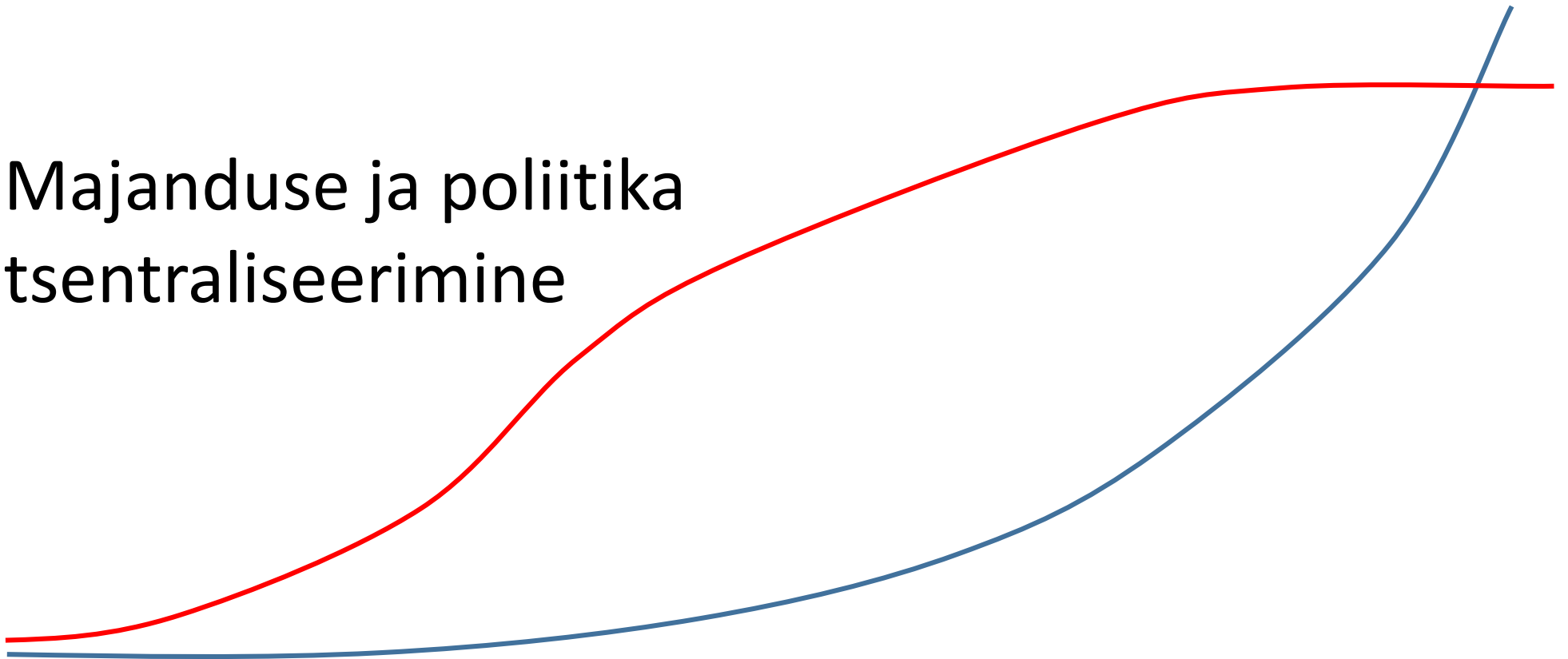
Erinevad suunad, eesmärgid, tehnoloogiad:

- Tsentraliseerimine
- Automatiseerimine
- Tehisintellekt



Vastastikune mõju
ja võimendus

Majanduse ja poliitika
tsentraliseerimine



Tööde automatiseerimine: AI ja
robotika